# Contributing

We would love for you to contribute to `ng-formly` and help make it even better than it is

today! As a contributor, here are the guidelines we would like you to follow:

- [Code of Conduct](#coc)

- [Question or Problem?](#question)

- [Issues and Bugs](#issue)

- [Feature Requests](#feature)

- [Submission Guidelines](#submit)

- [Coding Rules](#rules)

- [Commit Message Guidelines](#commit)

- [Contributing to Community](#community)

## <a name="coc"></a> Code of Conduct

Help us keep Angular open and inclusive. Please read and follow our [Code of Conduct][coc].

## <a name="question"></a> Got a Question or Problem?

Feel free to chat on [gitter](https://gitter.im/formly-js/ng2-formly) and raise questions on Stack overflow using `angular 2 formly` tag. We also support questions or problems from the [issues](https://github.com/ngx-formly/ngx-formly/issues)

## <a name="issue"></a> Found a Bug?

If you find a bug in the source code, you can help us by

[submitting an issue](#submit-issue) to our [GitHub Repository][github]. Even better, you can

[submit a Pull Request](#submit-pr) with a fix.

## <a name="feature"></a> Missing a Feature?

You can \*request\* a new feature by [submitting an issue](#submit-issue) to our GitHub

Repository. If you would like to \*implement\* a new feature, please submit an issue with

a proposal for your work first, to be sure that we can use it.

Please consider what kind of change it is:

\* For a \*\*Major Feature\*\*, first open an issue and outline your proposal so that it can be

discussed. This will also allow us to better coordinate our efforts, prevent duplication of work,

and help you to craft the change so that it is successfully accepted into the project.

\* \*\*Small Features\*\* can be crafted and directly [submitted as a Pull Request](#submit-pr).

## <a name="submit"></a> Submission Guidelines

### <a name="submit-issue"></a> Submitting an Issue

Before you submit an issue, please search the issue tracker, maybe an issue for your problem already exists and the discussion might inform you of workarounds readily available.

We want to fix all the issues as soon as possible, but before fixing a bug we need to reproduce and confirm it. In order to reproduce bugs we will systematically ask you to provide a minimal reproduction scenario using http://plnkr.co. Having a live, reproducible scenario gives us wealth of important information without going back & forth to you with additional questions like:

- version of Angular used

- 3rd-party libraries and their versions

- and most importantly - a use-case that fails

A minimal reproduce scenario using http://plnkr.co/ allows us to quickly confirm a bug (or point out coding problem) as well as confirm that we are fixing the right problem. If plunker is not a suitable way to demonstrate the problem (for example for issues related to our npm packaging), please create a standalone git repository demonstrating the problem.

We will be insisting on a minimal reproduce scenario in order to save maintainers time and ultimately be able to fix more bugs. Interestingly, from our experience users often find coding problems themselves while preparing a minimal plunk. We understand that sometimes it might be hard to extract essentials bits of code from a larger code-base but we really need to isolate the problem before we can fix it.

Unfortunately we are not able to investigate / fix bugs without a minimal reproduction, so if we don't hear back from you we are going to close an issue that don't have enough info to be reproduced.

You can file new issues by filling out our [new issue form](https://github.com/ngx-formly/ngx-formly/issues/new).

### <a name="submit-pr"></a> Submitting a Pull Request (PR)

Before you submit your Pull Request (PR) consider the following guidelines:

?????? ?\*\*Please follow [commit message conventions](#commit)\*\*

### Development

1. run `yarn`

2. run `npm run demo` or `yarn demo`

3. write tests & code in TS goodness :-)

4. run `git add .`

5. run `npm run commit` and follow the prompt (this ensures that your commit message follows [our conventions](#commit)).

6. push your changes

7. create a PR with a link to the original issue

8. wait patiently :-)

## <a name="rules"></a> Coding Rules

To ensure consistency throughout the source code, keep these rules in mind as you are working:

\* All features or bug fixes \*\*must be tested\*\* by one or more specs (unit-tests).

\* All public API methods \*\*must be documented\*\*. (Details TBC).

\* Follow the existing styles (we have an `.editorconfig` file)

\* Document your changes in the README (try to follow the convention you see in the rest of the file)

\* Create an example for the website that demonstrates your changes so people can see how your changes work

## <a name="commit"></a> Commit Message Guidelines

We have very precise rules over how our git commit messages can be formatted. This leads to \*\*more

readable messages\*\* that are easy to follow when looking through the \*\*project history\*\*. We follow [`Angular` project Commit Guidelines](https://github.com/angular/angular/blob/master/CONTRIBUTING.md#commit)

## <a name="community"></a> Contributing to community

- Create plugins!

- Write blog posts!

- Record screencasts

- Write examples. The website is driven by examples.

[coc]: https://github.com/ngx-formly/ngx-formly/blob/master/CODE\_OF\_CONDUCT.md

[github]: https://github.com/ngx-formly/ngx-formly